CLAIMS

1. A method for processing an extensible mark up language (XML) document comprising:

parsing the XML document into schema elements and data elements;

converting the schema elements into data type definition (DTD) objects;

validating the data elements using the DTD objects; and

if valid, constructing an in-memory tree representation of the XML document using the data elements.

2. The method of claim 1, wherein the converting comprises:

calling a method in a first application program interface (API); and

as a result of calling the first method, calling one or more methods in a second API to construct the DTD objects.

3. The method of claim 1, wherein the converting comprises referencing one or more tables that define the schema elements and associated functions for processing the schema elements.

4. A computer-readable medium having computer-executable instruction, which when executed by a computer, performs the method of claim 1.

5. A method for processing an extensible mark up language (XML) document comprising:

parsing XML data to produce a fragment having multiple elements;

calling a first application program interface (API) to construct nodes in an in-memory tree representation of the XML document for the elements; and

calling, in response to said calling the first API, a second API to construct data type definition (DTD) objects used in validating the elements.

6. The method of claim 5, further comprising referencing one or more tables to determine functions for processing the elements.

7. The method of claim 5, further comprising calling a third API to validate the elements using the DTD objects.

8. The method of claim 7, further comprising calling a fourth API to build an in-memory tree representation of the XLM document using validated elements.

9. A computer-readable medium having computer-executable instruction, which when executed by a computer, performs the method of claim 5.

10. A method for processing a parsed XML document, comprising:

validating whether elements from the parsed XML document belong to a schema;

if valid, creating one or more data structures for individual elements;

using the data structures to build nodes in an in-memory tree representation of the XML document; and

processing one or more attributes for the individual elements.

11. The method of claim 10, wherein the validating comprises determining whether the element exists in one or more tables defining the schema.

12. The method of claim 10, wherein the processing comprises determining whether the attributes exist in one or more tables for the schema and if found, calling associated functions for handling the attributes.

13. A computer-readable medium having computer-executable instruction, which when executed by a computer, performs the method of claim 10.

14. An architecture for processing an extensible mark up language (XML) document comprising:

a parser to parse the XML document into elements including schema elements and data elements;

a schema node factory, called by the parser, to handle calls to construct a node in an in-memory tree representation of the XML document for the elements; and

a schema builder, called by the schema node factory, to construct data type definition (DTD) objects used in validating the data elements.

15. The architecture of claim 14, wherein the schema builder utilizes one or more tables to process the elements, the tables containing information defining a schema for the XML data.

16. A computer implemented with the architecture of claim 14.

17. A client-server system, comprising:

a server;

a client connectable to the server to exchange extensible mark up language (XML) documents;

at least one of the client and the server implementing the architecture of claim 14.

18. A computer, comprising:

a memory;

a processor;

a network component to communicate with a remote server and to send and receive XML documents;

a parser, stored in memory and executed on the processor, to parse an XML document into elements including schema elements and data elements;

a schema node factory, stored in memory and executed on the processor, that is called by the parser to handle calls to construct a node in an in-memory tree representation of the XML document for the elements; and

a schema builder, stored in memory and executed on the processor, that is called by the schema node factory to construct data type definition (DTD) objects used in validating the data elements.

19. The computer of claim 18, further comprising one or more tables stored in memory, the tables containing information defining a schema for the XML data, the schema builder utilizing the tables to process the individual elements.

20. The computer of claim 18, further comprising a validation node factory, stored in memory and executed on the processor, to validate the data elements using the DTD objects constructed by the schema builder.

21. The computer of claim 20, further comprising a tree builder node factory, stored in memory and executed on the processor, to build an in-memory tree representation of the XML document using validated data elements.

22. A system for processing an extensible mark up language (XML) document comprising:

means for parsing the XML document into schema elements and data elements;

means for converting the schema elements into data type definition (DTD) objects;

means for validating the data elements using the DTD objects; and

if valid, means for constructing an in-memory tree representation of the XML document using the data elements.

23. A computer-readable medium having computer-executable instructions, which when executed on a computer, define an application program interface comprising callable methods for:

processing an element from a parsed extensible mark up language (XML) document by validating the element as belonging to a schema and if valid, creating one or more data structures for the element;

processing one or more attributes for the element; and

processing a text node from the parsed XML document to determine if text is valid.

24. The computer-readable medium of claim 23, wherein said processing the element comprises determining whether the element exists in one or more tables defining the schema and if found, creating the one or more data structures.

25. The computer-readable medium of claim 23, wherein the processing the attributes comprises determining whether the attributes exist in one or more tables for the schema and if found, calling associated functions for handling the attributes.

26. A data structure stored in a computer-readable medium and used in processing extensible mark up language (XML) data, comprising:

a root table containing information that defines a schema, the root table containing one or more references to one or more sub-tables;

an element types sub-table referenced by the root table, the element types sub-table listing types of elements in the schema;

an attribute types sub-table referenced by the root table, the attribute types sub-table listing types of attributes in the schema, corresponding functions for handling the attributes, and type descriptions; and

a function table referenced by at least the element types sub-table, the function table listing functions for processing the elements in the schema.

27. A unit for processing parsed XML data comprising:

the data structure of claim 26; and

a schema builder to utilize the tables in the data structure to build data type definition (DTD) objects, the DTD objects being used in validating the XML data.

28. In a system for processing an extensible mark up language (XML) document having schema elements and data elements, a schema builder comprising:

one or more tables containing information that defines a schema for the XML document; and

code to convert the scheme element to data type definition (DTD) objects using the tables, the data DTD objects being used in validating the data elements.